SAI internal peripheral

SAI internal peripheral

Stable: 21.01.2020 - 13:23 / Revision: 21.01.2020 - 13:21

Contents

1 Article purpose

The purpose of this article is to:

- briefly introduce the SAI peripheral and its main features
- indicate the level of security supported by this hardware block
- explain how each instance can be allocated to the three runtime contexts and linked to the corresponding software components
- explain how to configure the SAI peripheral.

2 Peripheral overview

The SAI (Serial Audio Interface) offers a wide set of audio protocols, such as: I2S standards (LSB or MSB-justified), PCM /DSP, TDM and S/PDIF. The SAI contains two independent audio sub-blocks. Each sub-block has its own clock generator and I/O line controller, and can be configured either as transmitter or receiver.

2.1 Features

Refer to STM32MP15 reference manuals for the complete feature list, and to the software components, introduced below, to see which features are implemented.

2.2 Security support

All the SAI instances are **non secure** peripherals.

3 Peripheral usage and associated software

3.1 Boot time

The SAI is not used at boot time.

3.2 Runtime

3.2.1 Overview

SAI instances can be allocated to:

- the Cortex-A7 non-secure for use in Linux with ALSA framework
- the Cortex-M4 for use in STM32Cube with STM32Cube SAI driver

Chapter #Peripheral assignment exposes which instance can be assigned to which context.

3.2.2 Software frameworks

Do	Peri	Software frameworks			Comment
mai n Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Audio	SAI		ALSA framework	STM32Cube SAI driver	

3.2.3 Peripheral configuration

The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, then manually completed (particularly for external peripherals), according to the information given in the corresponding software framework article.

3.2.3.1 Configuration in Cortex-A7 non-secure software

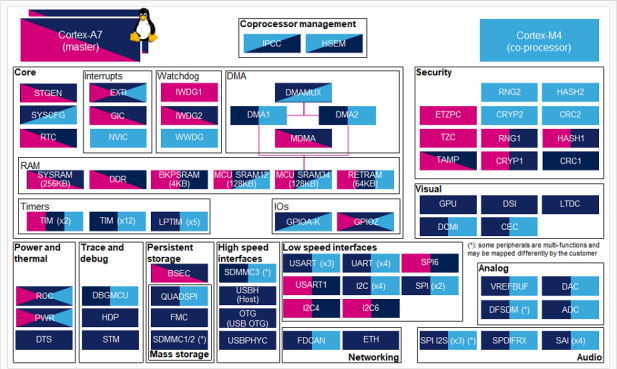
When the Arm® Cortex®-A7 core operates in non-secure access mode, the SAI is controlled by the Linux kernel framework. Refer to SAI Linux driver to drive the SAI through Linux kernel ALSA framework. Refer to Soundcard configuration and SAI device tree configuration to configure the SAI through the Linux kernel device tree[1].

3.2.3.2 Arm® Cortex®-M4 software configuration

3.2.4 Peripheral assignment

Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- means that the peripheral can be assigned () to the given runtime context.
- is used for system peripherals that cannot be unchecked because they are statically connected in the device.



Internal peripherals

Refer to How to assign an internal peripheral to a runtime context for more information on how to assign peripherals manually or via STM32CubeMX.
The present chapter describes STMicroelectronics recommendations or choice of implementation. Additional possiblities might be described in STM32MP15 reference manuals.

Domain	Peripheral	Runtime allocation				Comment
Instance		Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
		SAI1				Assignment (single

Domain	Peripheral	Runtime allocation				Comment
						choice)
Audio	SAI	SAI2				Assignment (single choice)
		SAI3				Assignment (single choice)
		SAI4				Assignment (single choice)

4 How to go further

STM32H7 SAI training [2] introduces the SAI features and applications. The SAI versions in STM32H7 and STM32MP15 are very close. In consequence this training is also relevant for STM32MP15. The user should refer to the STM32MP15 reference manuals for a complete description.

5 References

- Device tree
- STM32H7 SAI training

Serial Audio Interface (Mechanism used to transfer non-buffered audio data between processors and/or audio converters.)

Integrated Interchip Sound

Sony/Philips Digital Interface Format (Protocol (IEC-60958))

Open Portable Trusted Execution Environment